```
# Classification and Discriminant Analysis
# Exercise 9


# Load the data SAheart from the package ElemStatLearn.
# The data contain information about males in a heart-disease high-risk region
# in South Africa (see help). The goal is to apply logistic regression with
# natural cubic splines to split the data into groups according to the variable
# chd (coronary heart disease). Use the variable famhist in the exercise. Scale
# the data to have zero mean and unit variance.

library(ElemStatLearn)
data(SAheart)
# str(SAheart)
heart <- SAheart                                        # including the variable famhist
x.heart = heart[ , 1:9]         # everything except chd
y.heart = heart[ , 10]          # chd

heart$famhist <- as.numeric(heart$famhist)
x.heart.scaled <- scale(heart[ , 1:9])


# 1. k-means with prototypes: function kmeans()

# (a) Randomly select a training data set of 300 observations and apply k-means
# clustering for each group with 5 prototypes. Assign the test data with nearest
# prototypes to predict the group membership (classes of nearest prototypes) for
# the test data and compute the misclassification rate.

set.seed(123)
obs <- 300
train <- sample(1:nrow(heart), size = obs, replace = FALSE)

grp <- as.numeric(heart[ , 10])

mod.kmeans <- kmeans(x.heart.scaled[train, ], centers = 2, nstart = 5,
        algorithm = 'Hartigan-Wong')

TAB.kmeans <- table(mod.kmeans$cl, as.numeric(grp[train]))
TAB.kmeans
        #       0   1
        #   1 109  27
        #   2  87  77

mklrate.kmeans <- 1 - sum(diag(TAB.kmeans))/sum(TAB.kmeans)
mklrate.kmeans
        # [1] 0.38

# (b) Repeat the procedure 100 times and visualize the resulting
# missclassification rates with a boxplot.

set.seed(seed=NULL)
obs <- 300
proc <- 100
mklrate.kmeans.proc <- rep(0, times = proc)

grp <- as.numeric(heart[ , 10])

for (i in 1:proc){
```

```
        train <- sample(1:nrow(heart), size = obs, replace = FALSE)
        mod.kmeans <- kmeans(x.heart.scaled[train, ], centers = 2, nstart = 5,
                algorithm = 'Hartigan-Wong')
        TAB.kmeans <- table(mod.kmeans$cl, as.numeric(grp[train]))
        mklrate.kmeans.proc[i] <- 1 - sum(diag(TAB.kmeans))/sum(TAB.kmeans)
}

boxplot(mklrate.kmeans.proc, notch=TRUE, ylab='Rate of misclassification')
title(main='South African Hearth Disease Data
K-Means Clustering')


################################################################################


# 2. Learning Vector Quanitization: function lvq in library(class)

library(class)

# (a) Randomly select a training data set of 300 observations.

set.seed(123)
obs <- 300
train <- sample(1:nrow(heart),size = obs, replace = FALSE)

# Compute a codebook by using lvqinit() consisting of 10 prototypes.

prototyp <- lvqinit(x.heart.scaled[train, ], y.heart[train], size = 10)

# Now apply different algorithms (lvq1, lvq2, lvq3, olvq1),

mod.lvq1 <- lvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
mod.lvq1

mod.lvq2 <- lvq2(x.heart.scaled[train, ], y.heart[train], prototyp)
mod.lvq2

mod.lvq3 <- lvq3(x.heart.scaled[train, ], y.heart[train], prototyp)
mod.lvq3

mod.olvq1 <- olvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
mod.olvq1

# predict the group membership for the test data by using lvqtest and

predict.lvq1 <- lvqtest(mod.lvq1, x.heart.scaled[-train, ])

predict.lvq2 <- lvqtest(mod.lvq2, x.heart.scaled[-train, ])

predict.lvq3 <- lvqtest(mod.lvq3, x.heart.scaled[-train, ])

predict.olvq1 <- lvqtest(mod.olvq1, x.heart.scaled[-train, ])


# compute the misclassification rates.

TAB.lvq1 <- table(y.heart[-train], predict.lvq1)
TAB.lvq1
        #    predict.lvq1
        #      0  1
```

```
        #   0 95 11
        #   1 36 20

mklrate.lvq1 <- 1 - sum(diag(TAB.lvq1))/sum(TAB.lvq1)
mklrate.lvq1
        # [1] 0.2901235

TAB.lvq2 <- table(y.heart[-train], predict.lvq2)
TAB.lvq2
        #    predict.lvq2
        #      0  1
        #   0 95 11
        #   1 37 19

mklrate.lvq2 <- 1 - sum(diag(TAB.lvq2))/sum(TAB.lvq2)
mklrate.lvq2
        # [1] 0.2962963

TAB.lvq3 <- table(y.heart[-train], predict.lvq3)
TAB.lvq3
        #    predict.lvq3
        #      0  1
        #   0 97  9
        #   1 39 17

mklrate.lvq3 <- 1 - sum(diag(TAB.lvq3))/sum(TAB.lvq3)
mklrate.lvq3
        # [1] 0.2962963

TAB.olvq1 <- table(y.heart[-train], predict.olvq1)
TAB.olvq1
        #    predict.olvq1
        #      0  1
        #   0 93 13
        #   1 39 17

mklrate.olvq1 <- 1 - sum(diag(TAB.olvq1))/sum(TAB.olvq1)
mklrate.olvq1
        # [1] 0.3209877

# (b) Repeat the procedure 100 times and
# visualize the resulting missclassification rates with a boxplot.

set.seed(seed=NULL)
obs <- 300
proc <- 100
mklrate.lvq1.proc <- rep(0, times = proc)
mklrate.lvq2.proc <- rep(0, times = proc)
mklrate.lvq3.proc <- rep(0, times = proc)
mklrate.olvq1.proc <- rep(0, times = proc)

grp <- as.factor(y.heart)

for (i in 1:proc){
        train <- train <- sample(1:nrow(heart), size = obs, replace = FALSE)
        prototyp <- lvqinit(x.heart.scaled[train, ], y.heart[train], size = 10)
        mod.lvq1 <- lvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
        predict.lvq1 <- lvqtest(mod.lvq1, x.heart.scaled[-train, ])
        TAB.lvq1 <- table(y.heart[-train], predict.lvq1)
        mklrate.lvq1.proc[i] <- 1 - sum(diag(TAB.lvq1))/sum(TAB.lvq1)
```

```
        mod.lvq2 <- lvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
        predict.lvq2 <- lvqtest(mod.lvq2, x.heart.scaled[-train, ])
        TAB.lvq2 <- table(y.heart[-train], predict.lvq2)
        mklrate.lvq2.proc[i] <- 1 - sum(diag(TAB.lvq2))/sum(TAB.lvq2)
        mod.lvq3 <- lvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
        predict.lvq3 <- lvqtest(mod.lvq3, x.heart.scaled[-train, ])
        TAB.lvq3 <- table(y.heart[-train], predict.lvq3)
        mklrate.lvq3.proc[i] <- 1 - sum(diag(TAB.lvq3))/sum(TAB.lvq3)
        mod.olvq1 <- olvq1(x.heart.scaled[train, ], y.heart[train], prototyp)
        predict.olvq1 <- lvqtest(mod.olvq1, x.heart.scaled[-train, ])
        TAB.olvq1 <- table(y.heart[-train], predict.olvq1)
        mklrate.olvq1.proc[i] <- 1 - sum(diag(TAB.olvq1))/sum(TAB.olvq1)
}

boxplot(mklrate.lvq1.proc, mklrate.lvq2.proc, mklrate.lvq3.proc,
        mklrate.olvq1.proc, notch = TRUE, ylab = 'Rate of misclassification',
        names = c('lvq1', 'lvq2', 'lvq3', 'olvq1'))
title(main='South African Hearth Disease Data
Learning Vector Quantization')


################################################################################


# 3. knn classification: function knn in library(class)

library(class)
library(chemometrics)

# (a) Randomly select a training data set of 300 observations.

set.seed(123)
obs <- 300
train <- sample(1:nrow(heart), size = obs, replace = FALSE)

# Use the function knnEval from the library(chemometrics) to specify the optimal
# k (the number of nearest neighbors).

grp <- as.factor(y.heart)

resknn <-
        knnEval(X = x.heart.scaled, grp = grp, train = train,
                knnvec=seq(1, 30, by = 1), plotit = TRUE, legpos = "bottomright")
title("kNN classification")
indmin <- which.min(resknn$cvMean)
res <- resknn$cvMean[indmin] + resknn$cvSe[indmin]
fvec <- (resknn$cvMean < res)
indopt <- min((1:indmin)[fvec[1:indmin]])
indopt
        # [1] 15

# (b) Apply knn() on the training set with k from (a),
# predict the group membership for the test data and

mod.knn <- knn(train = x.heart.scaled[train, ], test = x.heart.scaled[-train, ],
        cl = grp[train], k = indopt)

# compute the misclassification rate.

TAB.knn <- table(y.heart[-train], mod.knn)
```

```
TAB.knn
        #    mod.knn
        #      0  1
        #   0 93 13
        #   1 35 21

mklrate.knn <- 1 - sum(diag(TAB.knn))/sum(TAB.knn)
mklrate.knn
        # [1] 0.2962963

# (b) Repeat the procedure 100 times and
# visualize the resulting missclassification rates with a boxplot.

set.seed(seed=NULL)
obs <- 300
proc <- 100
mklrate.knn.proc <- rep(0, times = proc)

grp <- as.factor(y.heart)

for (i in 1:proc){
        train <- train <- sample(1:nrow(heart), size = obs, replace = FALSE)
        resknn <-
                knnEval(X = x.heart.scaled, grp = grp, train = train,
                        knnvec=seq(1, 30, by = 1), plotit = FALSE)
        indmin <- which.min(resknn$cvMean)
        res <- resknn$cvMean[indmin] + resknn$cvSe[indmin]
        fvec <- (resknn$cvMean < res)
        indopt <- min((1:indmin)[fvec[1:indmin]])
        mod.knn <- knn(train = x.heart.scaled[train, ], test = x.heart.scaled[-train, ],
                cl = grp[train], k = indopt)
        TAB.knn <- table(y.heart[-train], mod.knn)
        mklrate.knn.proc[i] <- 1 - sum(diag(TAB.knn))/sum(TAB.knn)
}

boxplot(mklrate.knn.proc, notch=TRUE, ylab='Rate of misclassification')
title(main='South African Hearth Disease Data
k-Nearest Neighbour Classification')
```